BBC RD 1985/10

BBC

RESEARCH DEPARTMENT

REPORT

DIGITAL AUDIO EDITING

G.W. McNally, B.Sc., C.Eng., M.I.E.E., M.I.E.E.E.
P.S. Gaskell, M.A., C.Eng., M.I.E.E. A.J. Stirling, B.Sc., A.R.C.S.

Research Department, Engineering Division
THE BRITISH BROADCASTING CORPORATION

August, 1985

It uses SI units in accordance with B.S. document PD 5686.

BBC RD 1985/10

DIGITAL AUDIO EDITING

G.W. McNally, B.Sc., C.Eng., M.I.E.E., M.I.E.E.E.
P.S. Gaskell, M.A., C.Eng., M.I.E.E. A.J. Stirling, B.Sc., A.R.C.S.

Summary

New techniques are needed for reliable, fast and accurate editing of digital audio. The objective is to provide the facilities to which audio editors have grown accustomed while maintaining high operational speed and precise control, even when a tape is cut. Three levels of performance are discussed. The first and simplest is to cut the tape and use error concealment and electronic crossfading to smooth the splice. In a more advanced option, the concept of separate cut-point and edit-point is introduced using an auxiliary data track, termed a 'Labels' track, to control a 'jump' over the splice. In the third level, audio is transferred to direct access disc, either to assist in the rebearsal of tape-cut edits, simple or jump, or as a self-contained editor in which real-time, non-destructive editing is efficiently carried out.

The hardware and software of the editor is discussed at a system level with particular attention to the man-machine interface, the digital signal processing required for edit-point location, the role of the auxiliary labels data channel and the relationship of the editor with digital audio tape recorders.

Issued under the Authority of

M.E.B. Moffat

Head of Research Department

Research Department, Engineering Division
BRITISH BROADCASTING CORPORATION
August 1985
(PH-274)

DIGITAL AUDIO EDITING

G.W. McNally, B.Sc., C.Eng., M.I.E.E., M.I.E.E.E.
P.S. Gaskell, M.A., C.Eng., M.I.E.E., A.J. Stirling B.Sc., A.R.C.S.

DIGITAL AUDIO EDITING

G.W. McNally, B.Sc., C.Eng., M.I.E.E., M.I.E.E.E.
P.S. Gaskell, M.A., C.Eng., M.I.E.E., A.J. Stirling B.Sc., A.R.C.S.

1. Introduction

Audio editing operations and requirements in the recording and broadcasting industries are diverse, ranging from traditional cut-tape editing in radio and recording studios to various forms of dub editing for television and separate magnetic film sound recording. This Report reviews contrasting editing methods particularly in broadcasting, and explores the requirements of digital audio editing. To meet the demands of a wide range of potential applications, a hierarchy of editing strategies with varying degrees of sophistication is proposed.

The first and simplest is to cut the tape and use error concealment and electronic crossfading to smooth the splice. In a more advanced option, the concept of separate cut-point and edit-point is introduced, using an auxiliary data track to control a 'jump' over the splice.

The top level is a disc-based strategy which gives the user a flexible, non-destructive editing technique with advanced rehearsal facilities not possible with conventional methods. Based on this strategy, an experimental disc-based editor is being developed. The design philosophy and implementation of this editor are described together with a simulation of its performance. Particular attention is given to the man-machine interface, data formatting, and systems level design. The software engineering of the project is also reported.

2. Current editing practice

In order to understand more fully the breadth of editing operations, a number of studios were visisted in which different editing activities were practised depending on programme type. The more critical operations are reported in this Section.

2.1. News and current affairs

In a large broadcasting organistion, material for new and current affairs may originate in a number of ways. It may be,

1. Recorded in a studio locally with high quality.
2. Sent via lines from remote studios with high quality.
3. Brought in as pieces of ¼″ tape, average length 6-9 mins. with variable quality depending on the location and circumstances of the correspondent.

In the BBC, the storage requirements can be estimated from the amount of tape issued for the recording of local and remote studio output which is about 45 hours/day, and the amount of additional tape brought in at about 5 hours/day. After editing, this is reduced to a total of approximately 12 hours/day which may be stored for up to 6 weeks before being scrapped. Many items are sent via lines to the BBC's Local Radio Stations.

The requirements of an editing system are severe,

1. Multiple copies may be required if the material is used for more than one programme.
2. Last minute news items may result in material being broadcast almost as soon as it is brought in. Normally, however, the edited tape is physically carried to the appropriate studio for transmission.
3. A large proportion of the material is of high quality and the use of stereo is increasing.
4. Edit point location is often carried out at between two and four times normal tape speed in order to save time.

2.2. Drama

In the BBC, about 60% of drama is recorded in stereo, the remainder being in mono. Often, two parallel recordings are made – 'cold' and 'hot'. The 'cold' recording is of actors only and serves as a back-up to the 'hot' recording which is complete with effects, background noises, etc. In a way, this pre-empts the use of multichannel recording for drama.

A typical drama studio has six turntables and five two-track tape machines. The turntables are used for playing special effects discs and about 50 discs (5 hours) of preselected effects

material will be on-hand in the studio. Sometimes, a sub-mix of special effects will be prepared beforehand and dubbed to tape.

Drama recordings tend to be highly ordered, consisting of 'takes' lasting approximately 15 minutes and corresponding 're-takes' made in chronologically correct order. It is rare for more than three re-takes to be needed. For complex productions, special effects are added later indicating a trend toward separate recording and post-production sessions. However, this is tempered by the need for effects at the time of recording to provide ambience for the actors. (e.g. to cue actors so that they can shout above the noise of waves). Editing is required to join final takes, to remove 'ums' and 'ers', and to assemble the final copy. The long distance between edits and the relatively small number of edits suggest that a tape-cut editing approach may remain the best.

2.3. Classical music

Nearly all recordings are made in stereo. The desire for the highest possible quality, places some severe demands on every edit, and experience so far suggests:

1. Precise edit point location is a recurring problem. On occasions, a sharp pencil will be used to mark the edit rather than the more usual wax pencil, and this can be translated as a required resolution of better than 5ms.

2. Organs, horns and flutes are difficult to edit well and demand careful control of the cross-fade, i.e. the diagonal splice. Sometimes a 'chevron' splice is made to guarantee that both channels are cross-faded simultaneously.

3. Gain changes across the edit point are occasionally used. At the moment an extra dubbing is necessary to achieve this in analogue recordings and this may explain why it is not done more often.

A 'difficult' editing session may be summarised by the following example. An opera with re-takes was recorded on fourteen reels of tape and was reduced to five reels (two and a half hours) by editing. Three tape machines were used; the main takes on one, the re-takes on another, and a third for dubbing when gain changes were required. In this example, eight three-hour sessions were needed to create 140 edits, at least half of which involved two or three attempts. Thus the analogue tapes may be cut up to 400 times, two-thirds of which must be repaired to a very high standard. Even for a 'typical' editing session, about 20% of edits may have to be repaired.

2.4. Popular music

The recording of a popular music item begins with 'laying down' the main rhythm tracks on a multi-track machine, broadly on the basis of one track per microphone. Subsequent tracks (e.g. vocals) may be added synchronously but one pass may be sufficient to record the whole item. If a particular track is not satisfactory, it can be replaced by the artist 'overdubbing' that section. At the completion of the recording session, the 24-track tape consists of one version of the musical item (unwanted takes are discarded). The multi-track recorder therefore fulfills many of the editing requirements of a pop music studio.

The mixing stage is a rehearse/record process. The 24-track tape is replayed and the studio manager and producer mix the sound, adding equalisation, reverberation and special effects, perhaps in a computer-assisted mode, until the wanted sound is achieved. The tape may be replayed many times before the right effect is created. On the final replay, the desk output is recorded to ¼" inch stereo tape. If the item is very long or the mix is particularly complex, the final recording process may be split into a number of sections. These are then edited together. The edits tend to be straightforward since like is joined to like. In one example, a six-minute item was recorded with only one edit.

2.5. Features

Recorded material for a features programme is a mixture of music, interviews etc. Although almost entirely stereo, there is a trend towards multitrack operation because of the difficulties of maintaining high standards when double tracking, and the improvements in speed and efficiency when attempting, for example, to cue a special effect with an existing recording.

Editing work is characterised by a large number of edits. In interviews, particularly, 'ums' and 'ers' must be removed together with repetitions and redundant phrases. Several items must be assembled into a smoothly flowing programme, though it may be necessary to change the order of the programme at the very last minute. Edits can be as frequent as every two seconds whilst the total length of the material being edited

tends not to exceed 1-2 hours. Rough edits may be improved by the addition of 'breaths' between sections. These are often kept on one side after editing out from another portion of the tape. The 'breaths' do not have to be from the same speaker and sometimes not even the same sex.

Edit points are located by monitoring the tape at rewind speed (20 times normal) and a completed edit must be reviewed for at least 15 s. around the edit point in order to maintain an awareness of speaker rhythms. Shuttling the tape back and forth to find the various items contributes significantly to studio time.

Insert edits are frequently required and it is common to mix in a special effect over a short section without dubbing the entire tape. 'Clicks' and 'pops' from old records are currently removed by editing.

2.6. Film sound

As might be expected, the techniques for audio editing with film are quite different from the methods so far described. The European norm for film for television is for single camera shooting in which the camera is locked at 25 frames/s and audio is recorded on a portable recorder with a 50 Hz synchronising track.

Firstly, a cutting copy is made of the film and the picture editing is carried out. Only then is the audio editing started – a four stage process.

1. The Transfer Suite – The location audio tapes are dubbed on to 16mm magnetic film ('mag') using a synchroniser to lock the 50 Hz sync. track to the film sprocket holes. Several copies may be made and during this time, the picture film is processed and checked.

2. The Sync-up Room – Absolute synchronisation between film and sound is derived from the clapperboard. If the process has been successful so far, 'rubber numbers' are then coded on all sets of film.

3. The Cutting Room – Audio edits often correspond with picture edits, but the optimum edit point is unlikely to occur at identically the same place. The picture is normally replayed in synchronism with two audio mags. so that an overlap period can be arranged[1]. As the editing process is carried out, so the 'active' material switches from one mag. to another, unused portions being replaced by sprocketed blue leader or old film stock. An edit list is kept with footage counts to provide a record of the work. Special effects or background material may be dubbed on to further lengths of mag. stock and checked for synchronisation by manually replaying with the picture. Adjusting the relative timing between audio tracks is simply achieved by jumping sprocket holes. By this time, the cutting copies have been extensively handled, so the edit list is sent back to the transfer suite to produce a 'clean' version for dubbing.

4. The Dubbing Theatre – The clean mags. are replayed using several transports, a further two being used to record the output. These feed a mixer which is also sourced by ordinary tape and grams for additional effects. The pictures are replayed in the studio with a display of the footage counter and the producer crossfades between mags., cues effects etc., according to the key sheet for the final copy.

The final result can be no better than third generation and may be up to fifth generation. Typically there is one audio edit per four picture edits with an upper range of 200 audio edits in a 10 minute reel. Common defects such as creeping sync. (caused by failing camera batteries) can lead to 1 frame/s or 1 semi-tone error and are currently rectified by making very many small edits.

The major problem of film sound is that if the pictures are re-edited, then the whole process has to be repeated to generate a new edit list for the audio.

2.7. Video tape sound

A basic 1″ editing system consists of a source VTR, a record VTR, a simple mixing desk and a ¼″ audio tape recorder. Butt joins are achieved by 'parking' the source and record VTRs using time-code, starting them on a 10 second run-up and switching over electronically.

Cross-fade edits exploit the spare audio tracks on the record VTR. Pre-edit sound from the source VTR is dubbed to a spare audio track of the record VTR and allowed to run on so as to overlap the edit point. The two VTRs are then reparked, set into motion under time-code control, and at the edit point, the VT editor cross-fades between the pre-recorded sound on the record VTR and the source VTR, the resultant

sound being recorded on to the main audio track of the record VTR. The duration of the cross-fade can vary from near instantaneous to many seconds. This process is known as 'edit-smoothing'. All these operations are normally rehearsed before committing the sound to tape.

The ¼" machine is used to,

1. Add additional sound (e.g. music, sound effects, applause) to the record VTR.

2. Lift sound off the record VTR, edit it on ¼" tape (to remove spurious studio noise, say) and return it to the record VTR.

Synchronisation of the ¼" tape with the video tape is achieved by manually marking the ¼" tape and starting it from the appropriate record or replay head when recording or replaying. Synchronism is normally maintained for about 30 seconds with this method.

However, it is becoming more common for the sound to be sourced from a ¼" recorder with time-code on either the second audio track or a third centre track. As well as providing sound synchronous with vision, such a machine can also be used for edit smoothing in stereo, where phase integrity of the separate channels must be preserved.

Where specially written music or sound effects have to be added, or where a complex operatic sound is to be balanced, the above approach has been extended to synchronise multi-track audio recorders with a final post-production dub to the original master video-tape[2].

2.8. Summary of edit types

From the previous discussions on editing techniques, it can be seen that there are only a few different kinds of edit.

1. Insert edits – In multi-channel or stereo recording, portions of one or more existing channels are altered by inserting fresh material from the studio. The point at which 'punch-in' occurs must be accurately set up, preferably with the facility to rehearse.

2. Generation of master tracks – In multi-channel recording, two or more takes of the same material may be recorded on different channels. The desired portions are then recorded on an unused channel and this relieves channels for further material to be recorded. This can provide a useful intermediate stage in an editing process and allows the sound engineer to concentrate on other things during the final mixdown.[3]

3. Assembly edits – Linking wanted passages of speech or music from different takes or sections of the master recording.

4. Tightening-up – on a stereo master recording, it is often necessary to adjust playing time or remove quiet passages between verses, etc. For non-music recording it is common to remove sentences, words, 'ums' and 'ers', or a stutter.

5. Generation of album master – The final mixes of the various items on an album must be arranged in the desired sequence with the proper amount of lead-in between each selection.

6. Track synchronisation – Adjusting the relative timing of audio tracks relative to each other or to video/film.

3. Specification of a digital audio editor

3.1. Review of current methods for digital editing

There are already several differing strategies for editing digital audio (Fig. 1). The use of

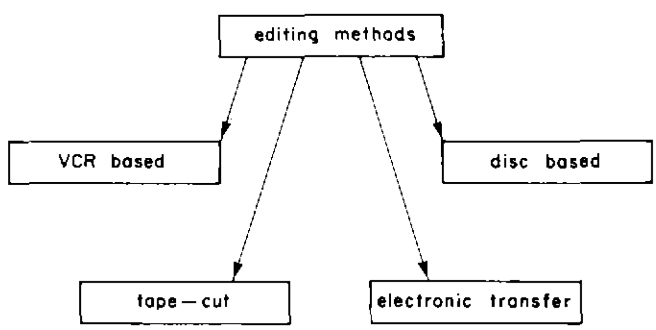


Fig. 1 – Digital audio editing methods.

helical-scan video-cassette recorders has grown in popularity as a low-cost medium for digital audio recording and a number of editing systems have been built around them[4]. As might be expected, the rotating heads and cassette format make dub-editing a necessity; it limits the facilities that can be provided and leads to time-consuming operations but offers an accuracy of 0.3ms in location. For stationary head, longitudinal, Digital

Audio Tape Recorders (DATRs), electronic editors have been developed for multichannel audio[3]. Although still relying on dubbing, new facilities were introduced such as compiling a single master track from a number of takes on other tracks, or 'punching' in and out while retaining digital quality. For stereo work (and sometimes multichannel), cut-tape methods have been introduced[5]. These present difficulties in rehearsing an edit before cutting a valuable master tape, and concealing the corrupted data at the splice. Finally, a number of off-line systems have been developed in which audio material is dubbed to a disc-drive and edited via a mainframe computer[6,7,8]. Many facilities are provided such as edit rehearsal and audition but the systems are costly and are not usually installed at the recording studio.

These examples indicate that there is a useful role for both direct-access type editors and linear, sequential access types, that is until a removeable, bulk random access storage is cost effective for, say, 2 hours of stereo recording. For this reason, we have proposed a hierarchy of editing strategies which are to a great degree mutually compatible and provide a range of facilities depending on the level in the hierarchy.

3.2. General requirements for an advanced digital editor

An advanced digital audio editor should be capable of providing solutions to all the problems highlighted in Section 2, though any one implementation may only attempt a specialist solution. In addition, the editor must adapt to new developments in digital audio for which a full discussion would be inappropriate and premature. However the following features should be considered.

1. Storage: A few minutes of storage may suffice for cueing a small number of effects, but a requirement for 1-2 hours would be likely for drama, classical music and features applications. This must be available on-line using discs for direct access. In addition, bulk removable storage will be needed and this could be satisfied by conventional DATR, magnetic data cartridge or optical disc.

2. Edit point location: Most editing is assessed by listening and most applications would benefit from features such as rock-and-roll and monitoring at spooling speeds.

3. Auxiliary data : Digital audio signals can be accompanied by other data which may be for operational use at the studio or intended for consumer consumption. This must be edited in many cases with the audio, e.g. Compact Disc sub-code data.

4. Administrative help : Complex editing may result in long edit decision lists or many takes may be simultaneously available on the system. In either case full documentary and cataloguing support would be needed.

5. Edit precision : Editing resolution in the region of 1ms is necessary with full control over cross-fade duration. Gain change in the vicinity of the edit is also desirable. Full rehearsal facilities are assumed.

6. Interfacing : The edit must fit into planned studio organisation, e.g. it may connect to SMPTE remote control of DATRs, time-code synchronisers, and even to synthesisers.

7. Man-machine interface : This would be an area of continuous development. It must therefore be carefully structured and based on software written in a high level language.

4. Digital tape-cut editing

4.1. Basic tape-cut editing — Level 1

In a similar way to analogue editing, the edit points on the tape are located by rocking the tape back and forth across the edit point ('rock and roll') using an analogue cue track, the tape is then cut and spliced. Data in the immediate vicinity of the edit is corrupted and, on replay, the decoding circuitry invokes an error concealment and cross-fade strategy to provide an acceptable edit. At least two commercially available DATRs have this form of editing although the error detection and concealment methods are different. It is a rapid editing technique and for many purposes, the quality of edit is satisfactory. Under critical conditions, however, impairments at the edit are occasionally audible. The quality of the cue track is also sometimes a limitation and, of course, it is not possible to rehearse the edit.

4.2. 'Jump' tape-cut editing — Level 2

A strategy has been devised that allows 'perfect' electronic edits to be performed on tape in conjunction with cutting and splicing. The notion of separate cut and edit points is introduced[9,10]. The edit points are displaced respectively

to the left and right of the cut points on the lead-in and lead-out sections (see Fig. 2). Data relevant to the edit may then be decoded and cross-faded without errors by 'jumping' over the corrupted

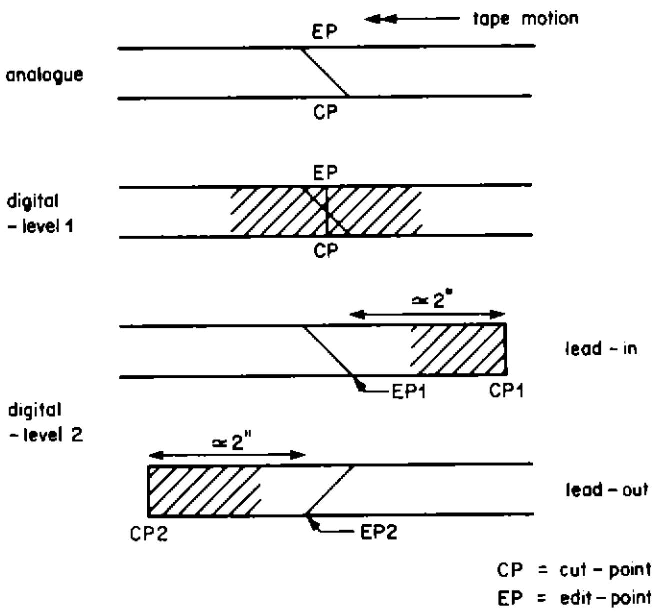


Fig. 2 — Tape-cut editing strategies.

data at the cut point. The displacement of the edit points and the overlap of the data leads to a gap in the audio data stream. This is smoothed by a buffer which is replenished by increasing the tape speed after the edit, an additional function for the servo-loop already present in all digital recorders.

Additional data must be written on the tape to indicate the location of the edit points and the cross-fade parameters to be used. This data may conveniently be packaged as 'Labels' which are discussed later in this Report.

This editing strategy not only allows electronic edits to be performed but is compatible with the basic, Level 1 edit. Thus, an edit in the absence of any Labels is correctly decoded as a Level 1 edit. However, if a Level 2 edit is replayed on a recorder without the appropriate Labels and store hardware, then the edit is replayed with error concealment but displaced in time from its correct position (typically by ±0.3s. at 7½ inches/s (ips)). This timing discrepancy is only likely to be acceptable under non-critical conditions (such as during silent passages).

Level 2 editing is entirely DATR based and rehearsal facilities are necessarily limited; the edit points (and hence the cut points) are found with the aid of the cue track and 'rock and roll' manipulation — as for Level 1 editing. If rehearsal facilities are needed then a Level 3 strategy should be used.

5. Disc assisted tape-cut editing — Level 3

The direct access nature of a hard disc can be exploited to give the user extensive rehearsal and editing facilities. Real-time and off-line signal processing can be introduced and this opens the door to a host of production facilities that are impossible with a tape-only strategy.

In the context of tape-cut editing, the disc-based editor is a peripheral to the DATR. Short sections (30s., say) of audio are dubbed to the disc where the edit is rehearsed. The edit information, including the location of the cut point, is then transferred back to tape. When this operation is completed, the edit on the tape is a Level 2, jump and matches precisely that rehearsed on disc. To replay the edit, the recorder requires only the same hardware as that for a Level 2 edit; the disc is no longer needed.

6. Disc based editing

A disc-based editor may be used not only as a peripheral to a DATR, but also as a totally self-contained mastering and editing facility. A practical system for studio use would require storage for at least one hour of stereo on a low-cost removable medium, and this is becoming feasible in the light of developments in high density magnetic and magneto-optic recording[11]. 'Write-once' optical discs[12] are now available with suitable computer interfaces and could find applications for archiving material. High cost implementations have already been constructed, for example, by Tokyo Broadcasting for scheduling commercials[13].

6.1. Factors determining editing performance

The performance of the system depends critically on the dynamic characteristics of the disc, on its controller, on the method of data buffering and on the sampling rate and number of channels of audio to be handled. A Winchester disc drive has several active surfaces each of which may have more than one head. Data is formatted on the surfaces into cylinders, tracks and sectors. The disc drive parameters relate to the time it takes for the heads to 'seek' a new sector and include the latency time (period of one revolution), track to track time and full sweep (maximum seek time for a new track).

The disc controller also plays a key role. Important features include its ability to transfer a track of corrected data at a single pass and to minimise the gap in data transfer at track boundaries by 'spiralling'.

Spiralling represents the degree to which data can be transferred without interruption and several modes can be identified. The most rapid mode occurs when the sector seek at a track boundary is sufficiently fast to permit continuous data transfer across the boundary without incurring a track latency delay. This is possible for multi-head drives when the heads can be electrically switched within cylinders.

However, under normal circumstances, when a seek to an adjacent cylinder is initiated, the access time caused by head movement (typically 2-10ms) will incur a latency. Skew sectoring is a technique by which the first sector on each cylinder is offset to anticipate this delay, and can be used to great advantage in disc systems with many tracks of low capacity.

The factors determining the net transfer rates are therefore numerous and complex. A simulation program was developed to estimate the performance and to determine the required design parameters. It was sufficiently flexible to cater for a wide range of disc storage media and audio standards. It estimates the number of consecutive edits that may be performed as audio is replayed off disc, assuming worst case conditions. Fig. 3 shows the expected performance of the experimental editor. Up to three edits a second may be executed continuously for a cross-fade period

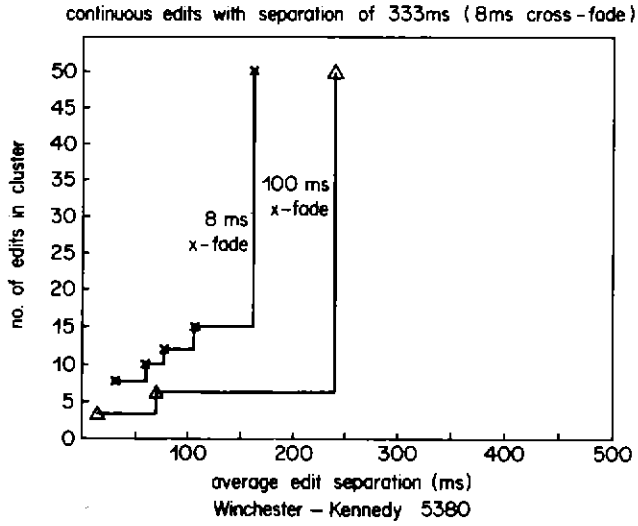


Fig. 3 — Simulation of editing performance.

of 8ms (equivalent to a 45° cut on ¼" analogue tape at 15 ips). If the separation between edits is reduced, clusters of edits are possible followed by a recovery period. Further simulations showed that, for example, a cluster of eight edits separated by 35ms must be followed by an edit free period of 1.4s. before further edits. Fig. 3 also shows how the number of possible edits is reduced if the crossfade period is increased to 100ms. This performance is expected to meet all but the most demanding situations, when a dub editing procedure may have to be used.

The simulation also showed that it was essential that the disc controller should be able to read data from a track with error correction at a single pass. At track boundaries, it was assumed that, within cylinders, the spiralling is sufficiently rapid to allow the uninterrupted transfer of data but that, at cylinder boundaries, a full track latency delay is incurred.

6.2. Defect handling

As storage densities increase, so the number of media defects also tend to increase. Permanent defects, which would cause data errors, must either be masked (i.e. that area of the disc surface is not used), or error correction techniques must be applied.

The technique of masking defects requires an identification of defects when the disc is formatted, and the mapping of replacement storage from other parts of the disc. This is undesirable because it can break up contiguous files and cause additional seeks, so lowering net data transfer rates.

During manufacture, the size of defects is kept under close control and this permits error correction to be used reliably. Though slightly more costly, a disc controller equipped with error correction can present a logically perfect disc to the host with the minimum perturbation to programs running in the host. The techniques are also being applied to optical discs (where the defects result in the loss of much larger numbers of bits) and it therefore appears to be a good approach for audio data transfers. Note that error correction schemes developed for DATRs are not entirely appropriate because of the need to format the disc.

6.3. Edit point location and cueing

Most sound editing and cueing can only be satisfactorily monitored by listening. An edit

point, for example, is conventionally found by first 'spooling' at high speed to the approximate location, then more accurately by 'rock-and-roll'. These are effectively variable speed operations over a range from zero to perhaps thirty times normal speed (X30). Digital audio tape recorders cannot currently reproduce successfully over such a speed range and are fitted with separate analogue tracks to assist in editing. An auto-locator may be used, based on time-code, but this is of more value when the sound is an accompaniment to video or film. In a disc-based audio recording and editing system, the data reading speed at the heads is constant, and permits a variety of techniques to be used to enhance the ease and quality of location and cueing.

6.3.1. Current disc-based techniques

The high data reading speed of magnetic disc systems (10Mb/s in medium cost units) may be exploited to give a four times replay speed, (X4), of stereo signals and, if the data is suitably organised, X8 for a single channel. However, this results in a varying data rate digital signal, and even at X8, falls short of operational requirements.

A second technique which has been demonstrated on the Compact Disc replays short excerpts at normal speed while 'skipping' through the material. The subjective effect of listening to the interrupted material is rather distracting, and of course the desired 'event' may be missed.

Bi-directional replay at low speeds, rock-and-roll, has been reproduced digitally by transferring a fixed amount of data into semiconductor storage and subsequently processing it[4]. However, such an approach lacks the freedom of searching through much more than approximately 10s. of material before a further 'dump' is needed with attendant delays.

6.3.2. New disc-based techniques

Two new methods are under investigation for fast edit point location and cueing. At the heart of the approach are the use of a data buffer to provide a regular and fully variable replay data rate from disc and signal processing to give controlled signal bandwidth and constant output sampling rate. These variable speed relay techniques are described in greater detail elsewhere[14,15].

The first method for fast edit point location involves pre-processing the audio during recording. A 'spooling-file' is created which is a reduced bandwidth, reduced sampling rate version of the original audio signal. This low data rate signal can then be replayed at much high speed before the data transfer rate from disc becomes a limitation. Since the file is still an audio file, it can be replayed through varispeed signal processing to give a very large range of speed control. The same data can be efficiently processed to provide a waveform display which may be useful in some applications.

The second method involves the participation of a sound editor or assistant during recording to generate 'edit markers' at relevant times, which can be augmented later during post production. The edit markers are logged in an auxiliary data file which contains other audio related data, as well as a simple time-coded list for fast subsequent retrieval. Editing may be further assisted by the use of formatted users' data or 'Labels', [16] which could be the script or score of the material coded in the auxiliary data file.

These methods are again discussed in detail elsewhere[17], but are mentioned here to indicate the demands that are made on the real-time data transfer systems.

6.4. The disc format and file organisation

It is convenient to adhere to conventional disc formatting into sectors, tracks and cylinders so that commercially available disc controllers can be used. For efficient use of the available storage capacity, a 'scatter storage' scheme should be used. Such schemes distribute a given file within a storage area as a set of discrete blocks of data which may be as small as a sector or as large as several tracks[18]. It has the great advantage that as material is deleted and recorded the full storage capacity can be retained even when the disc becomes chequer-boarded.

However, for audio transfers, the use of contiguous files gives significant advantages. Firstly, it guarantees the performance for high speed replay, i.e. it avoids unnecessary seeks when there is no editing. Secondly, it gives a direct relationship between time-code and disc address.

It has been indicated that a spooling file and an auxiliary data file are associated with each audio file. These should also be held in contiguous areas of the disc to avoid unnecessary chequer-boarding, and for a given time-code a pointer to each file is easily generated.

In contrast, directory information, edit decision lists, (edit files), and other system information is held in a separate reserved area of the disc. This area is maintained by the host operating system and contains all the system software including

the editing software, microcode for the real-time audio data transfers and programs for 'booting' the system, i.e. loading all necessary software when the system is turned on.

7. Real-time data interface

7.1. Requirements

A special purpose, high speed interface, called RIO (real-time input/output) has been designed to convert and process audio and auxiliary data between the computer bus and Winchester disc on the one hand, and the digital audio studio interconnections on the other. The principal requirements of this interface are as follows:

a) Data retiming: Data transfers between RIO and the Winchester disc are under DMA control and occur in high speed bursts. Each transfer has to be set up by the CPU and the disc heads have to 'seek' the start sector of the disc before the transfer can begin. This leads to a short gap between bursts even when the data is contiguous on the disc. It is advantageous therefore to make the DMA transfers as long as possible (in this case 64Kbytes). When the audio is edited, further delays are introduced depending on the location of the audio data on the disc. RIO contains a large memory buffer to retime the data to conform to the regular timing structure of the AES/EBU or other (e.g. parallel) standard.

b) Data formatting: For many applications, it is desirable to record auxiliary data, as well as the audio data on disc. This data may include labels, status, validity flags and ranging data from the AES/EBU link together with spooling data for high speed replay (see Sections 8 and 9). RIO assembles the different categories of data, so that they can be transferred to the appropriate files on disc. On replay, RIO performs the complementary reformatting process for transmission via the AES/EBU link.

c) Editing: When the audio on disc is edited, RIO fulfils several tasks. It performs a cross-fade (of user-defined duration) between the relevant audio passages. If requested, it introduces a gain offset across the edit that is controlled by a fader setting. After the edit, the gain may be restored to unity, again under fader control. Lastly, label and other auxiliary data are switched at (or close to) the edit point and RIO ensures that the integrity of the associated formats is preserved, reblocking if necessary (see Section 9.1).

d) Search and cueing operations : Section 6.3 discusses how edit points can be found rapidly by spooling and 'rock-and-roll'. During these operations, the audio is replayed in varispeed mode and RIO behaves as a demand-fed buffer, responding continuously to variable sampling rates. RIO also controls the replay of audio in the forward and reverse directions.

A common operational procedure associated with cueing is the instant start and stop of audio replay. Instant start is realised by pre-charging the data memory in RIO and issuing a start command via the edit control panel and the SYNC bus (see Section 12). Instant stop of the audio is similarly performed via the SYNC bus. A fade-up or fade-down is automatically carried out with start and stop commands.

7.2. Hardware

To fulfil the functions described in Section 7.1, RIO has to operate at high speed for real-time performance and it has to be programmable for versatility. The high speed Am29116 controller was chosen therefore as the control processor and, together with the other components of RIO it is microprogrammed. RIO is modular so that it is not restricted to any one computer system but can be integrated into future developments. To this end, a secondary interface links RIO to the relevant system bus (see Fig. 4). At the audio port of RIO,

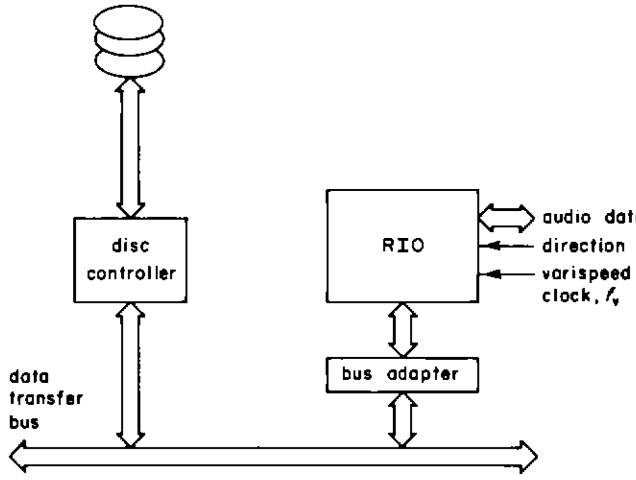


Fig. 4 — RIO as an audio peripheral.

the data is in 16-bit parallel form, suitable for linking to an AES/EBU transmitter/receiver, to an ADC/DAC codec or to a signal processor for further signal conditioning.

7.2.1. Data memory

A block diagram of the principal components of RIO is shown in Fig. 5. The most important feature is the large data memory, consisting of four segments each of 64Kbytes. This configuration was based on the results of the performance simulation, the maximum length of the DMA transfers and the configuration of the available 64Kbit dynamic RAMs. The memory is built into RIO so that it can be used with peripheral busses which do not support separate memory.

The memory segments may be individually switched, so that whilst one segment is used internally by RIO, another is concurrently available for DMA from the Winchester controller. Two high speed sequencers, operating at 50 MHz, provide independent timing signals for the two segments without any wait states. The segments are configured as 64K by 8-bit words and two read or write cycles are needed for each 16-bit word transfer, yielding a cycle time of 640ns. This matched the speed of available Winchester DMA controllers.

7.2.2. CPU

This section briefly describes the other components in RIO and how they fit into the overall architecture. The Am29116 was developed for high speed, complex control operations. Its word length is 16 bits and it offers a comprehensive selection of bit manipulations and shift operations, making it well suited to controlling RIO. It has a 32-word register file which can be addressed either by RIO itself or by the host computer. Certain registers are allocated for exchanging control information between RIO and the system computer. This is the main vehicle by which the host computer passes commands to RIO and by which RIO requests action by the host computer (e.g. to initiate a DMA transfer). Signal processing on RIO is furnished by a 16 x 16 multiplier (Am29517) and a 2K word coefficient memory. RIO is buffered from the real-time audio port by two 64-word FIFO's allowing simultaneous, bi-directional transfer of audio at any chosen sampling frequency. It permits RIO to operate with an independent system clock and it relaxes the timing constraints of the software. The wordwidth of the FIFO is 20 bits, providing 4 extra bits for synchronisation purposes, such as the alignment of the audio data with the block structure of the AES/EBU format. Auxiliary data is grouped into 16-bit words and multiplexed with the audio data for passage through the FIFO.

The architecture of RIO is shown in Fig. 5

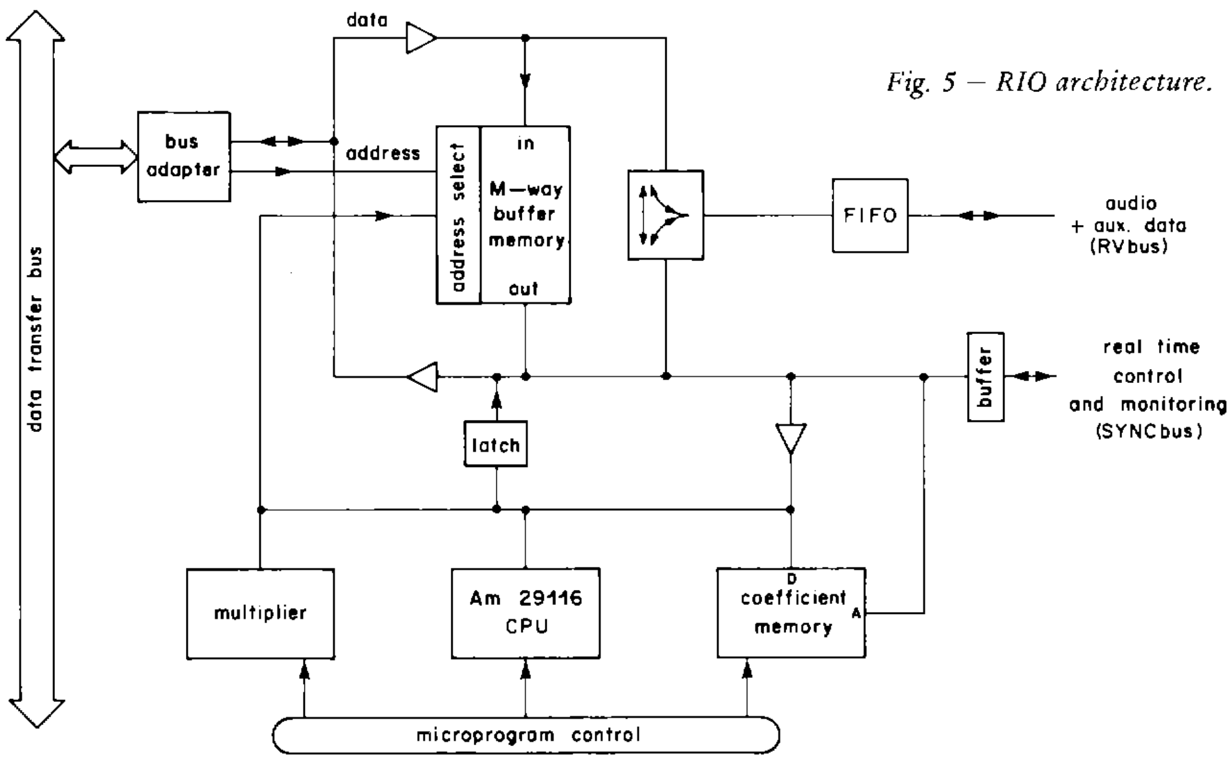


Fig. 5 — RIO architecture.

and is based on three busses, DTIN, DTOUT and Y which link the processing and storage components and the various input/output ports. All the buses are 16-bit wide. Buffers and latches at strategic positions isolate the appropriate bus so that the data memory can be accessed simultaneously by RIO and by the host computer bus.

As well as ports to the audio connections and to the system bus, there is an additional port to the SYNC bus for real-time user control from the edit control panel. It allows the host computer to monitor accurately the activity in RIO. This is useful during complex operations such as varispeed replay and updating time-code referenced to the inputs/outputs.

7.2.3. Microcode memory

The operation of RIO is controlled by 64-bit wide micro-instructions stored in a (1K x 64) RAM. The instruction fetch and execution cycles are pipe-lined giving a net execution rate of 6.25 MHz. Program flow is determined by an Am2910 sequencer, together with a mapping register for direct program control by the CPU.

Programs are stored on Winchester disc and, on switch-on, are loaded into the program memory of RIO. This is done by means of a microprogram editor run by the host computer. The editor also lists, edits and saves programs resident in RIO. This necessarily operates at the machine code level; for software development, a meta-assembler was used running on a VAX 11/750. This enabled the micro-code mnemonics to be defined and programs to be compiled and downloaded to the editor.

7.3. Software

The programs executed by RIO divide into a small number of tasks of which the principle ones are to record audio (and auxiliary data) on to disc and to replay data from disc. The record and replay tasks share a number of common functions such as memory refresh, memory segment management and command/status exchange with the host computer. The software is structured and the common functions are accessible as subroutines.

The CPU has 32 registers which are used for program control, for servicing memory and control registers and for storing regularly accessed variables. Three registers are shared by RIO and the host computer so that command and status information can be exchanged. These include a vector address for new commands, status information on RIO and a number of one-line handshake bits for data exchanges. Data which does not need immediate access is stored in the separate coefficient memory.

7.3.1. Real-time multi-tasking

The record and replay programs have to perform a number of time-critical functions including those directly related to the real-time nature of digital audio. As well as performing the main program, RIO ensures that each of these routines is executed within the time permitted. This is done by regularly polling the status of the particular function and servicing it as necessary. Polling represents a negligible overhead since, in most cases, it can be incorporated into a micro-instruction already in use for another purpose. The time-critical functions are as follows:

a) FIFO memory maintenance : During audio replay (recording), the FIFO must be maintained as full (empty) as possible. The FIFO flags that its input (output) register is ready and RIO responds by writing (reading) a word of data. For normal operation, the average response time should be less than 10μs.

b) Demand-fed buffering : During record and replay of audio, the data memory of RIO is used as a demand-fed buffer. One block is normally being accessed by DMA and the other three are maintained as full as possible by RIO. To this end, it is important to minimise the delay at the end of each DMA transfer, shown by a status flag in one of the command/status registers in the CPU. This flag is regularly monitored and if the end of a DMA transfer is indicated, the memory block is incremented ready for the next transfer. The average associated delay is about 80μs.

c) Command/status data exchange : Command and status information is exchanged between the host computer and RIO via shared registers. For the particular host computer (LSI 11/23) used for the prototype editor, a read/write cycle of the LSI 11/23 must be completed within 10μs to avoid a time-out trap. Such data exchanges involve program participation by RIO and requests for them by the LSI 11/23 have to be acknowledged within the period of 10μs.

d) DRAM refresh : The data memory in RIO

has to be refreshed every 2ms and this is performed by the CPU under software control. When a refresh time-out occurs, RIO must respond within a short period (of the order of 100μs) and execute the memory refresh routine.

7.3.2. Edit control blocks

When audio is edited, audio data corresponding to the lead-in and lead-out passages, including the edit period itself, are transferred to RIO data memory. There is a delay before the edit data is cross-faded and output to the audio port because of earlier audio data in the memory. Thus, a means is needed to store details of the edit so that when the time arrives for the edit to be performed, RIO can recall the details and execute the edit as instructed. To this end, when the audio edit data is transferred, additional control information in the form of an 'Edit control block' (ECB) is written to a reserved region of the current DMA memory block. The ECB gives precise information on the location, cross-fade time, gain across the edit and offset. When RIO comes to read a memory block, it first examines the reserved sector of the memory for a valid ECB. Any one memory block may include many ECB's (up to 64 per block). When one edit has been performed, the next ECB is recalled. ECB's are also used to fade-up and fade-down the audio at the start and end of a passage for a smooth start-up and stop.

8. The role of auxiliary data in editing

Auxiliary data is defined here as data carried by the AES/EBU serial interface[19] , other than the 16-bit audio data. Depending on the application, the auxiliary data can be as important as the audio data itself and therefore a means must be provided for its storage and processing. Auxiliary data is made up from:

a) Users' data: A 96Kbit/s channel for free format data. The data may however be formatted and one proposal than describes the data as 'Labels', which may be used for the text of a script or a musical score. The editor must edit this data with the audio and could usefully provide the tools to generate and modify such data.

b) Status data : A further 96Kbit/s channel for a variety of information, predominantly concerned with defining technical parameters of the audio signal. It contains two time-codes (more accurately, sample address codes) which provide the opportunity to preserve the original time-code through editing operations.

c) Validity and parity : Two 96Kbit/s channels which identify transmission errors and provide evidence of prior use of error concealment. The application and need for such information remain to be discussed.

d) Ranging data : The AES/EBU link provides for 24-bit audio, though most digital audio recording systems provide for only 16. A low capacity allocation for ranging data would provide a means for increasing the effective dynamic range of the recorded 16-bits. Clearly, new representations for audio data will make their own demands on the way editing is managed.

In addition, the editor may require some data capacity for its own use. Such an example would be the use of 'edit markers' which reveal the editing history of the material as it is replayed. Information such as 'bad note' does not fit neatly with either the concept of users' data or status and it is unlikely that all the data would be preserved through to the final output. However, such a technique permits greater freedom in locating and identifying audio material and reduces the reliance on time-code with its distracting emphasis on numbers.

In the current phase of the editing project, only users' data is stored at the full 96Kbit/s rate. This has been used to store the lyrics of a song so that editing can be assisted by viewing the text on a separate terminal. The audio and text are automatically edited together. The approach will be extended to incorporate all the auxiliary data mentioned above.

9. File formats

9.1. File structures and the AES/EBU interface

There are many advantages in configuring the audio, auxiliary data and other audio-related files so that there is a defined relationship with the timing of an original digital signal on the AES/EBU interface. Particularly relevant is the block structure of the interface identified by the BSYNC preamble[19] , and its relationship to the number of disc sectors used to store audio, auxiliary data, etc. It is then useful to think in terms of 'recording units', here defined as the minimum time interval in which an integer number of BSYNCs, audio sectors and auxiliary data sectors occur on disc. The objective is to simplify disc addressing and preserve the original block structure

of the audio in the edited output.

For example, if editing is to a resolution of 4ms, status data in the auxiliary data file is never sub-divided. However, if sample resolution is required, a strategy must be devised for handling the truncated blocks of status, users' data etc.

The experimental equipment permits reblocking, i.e. transferring the auxiliary data from one block phase to another, but the preferred approach is to edit without reblocking. This has the consequence that at an edit, there will be invalid auxiliary data and, in general, a shift in block phase. This should have no repercussions on other digital equipment using the signals and is a useful strategy for the general problem of handling auxiliary data at a switch, edit or synchronisation process.

The short interruption to the validity of auxiliary data should be identified on the AES/EBU interface. It is proposed that the status channel will carry a flag warning of the approach of invalid data, but no detailed strategy has yet been considered.

9.2. Audio files

Two methods were considered for the formatting of digital audio within a file. The first is suited to multiple channel working in which ready access to individual channels is required. In this case, a block of data from each channel is collected and then individually transferred to disc. The block, termed an allocation unit, is designed to correspond with a convenient amount of disc storage such as a complete track or multiple thereof. This makes the identification of the storage areas on disc corresponding to a particular audio channel relatively easy to administer. This method can also be used with scatter storage techniques to optimise storage availability[18]. It also permits features such as a delay of one channel relative to others to be achieved with ease. However, much larger data buffers are needed to smooth the interruptions of data flow, particularly at edits.

As stereo editing constitutes the major application area for an editor, a second method was adopted. A sample-by-sample multiplex of the two audio signals has the advantage that a single disc access before and after an edit produces all the necessary data. The penalty is that individual channels are inefficiently transferred and relative delays between channels require special purpose, though simple, output processing.

9.3. Files for searching and cueing

Provision has been made for the generation of two additional files with an audio recording – the spooling and the auxiliary data file. The spooling file must have the identical format to the audio file so that it can be replayed through a variable speed processor without reprogramming for a different format. The auxiliary data file is a completely different format and currently consists of users' data only, having a capacity of one sixteenth of the corresponding audio file. A general format for this file is being developed,[17].

10. Signal processing

As has already been indicated, the need for edit point location, cueing and signal monitoring at high speed requires special signal processing. A dedicated processor is under development to provide these variable speed operations and will provide several functions.

10.1. Edit point location and cueing

The output rate of data is derived from a position control knob designed to represent the spools of a tape recorder. Rotating the knob generates a variable rate clock and directional information so that RIO, acting as a demand-fed buffer, reads data from the disc in proportion to the rotation of the knob.

The variable speed processor (VSP) must remove the repeated spectra which will move through the audio band as the speed varies. The processed data may then either be resampled at a fixed rate for output to a digital system, or passed to a variable rate DAC[14,15]. The quality of the replayed signals will, by definition, be inferior to the full bandwidth, normal speed audio and so the processing accuracies can be relaxed slightly. The VSP under development will use a single chip signal processor, TMS32010, to carry out this processing.

10.2. High speed monitoring

The VSP will also be used while recording to generate a reduced bandwidth, reduced data rate version of the input for spooling purposes. Again, the filtering problem is one of removal of spectral repetitions before re-sampling at the reduced rate. A reduction to one sixteenth of the input sample rate is planned so that on replay, the audio can be played back via the VSP to give intelligible signals over a very wide speed range.[17]

11. The man-machine interface

11.1. Devices for display and input

11.1.1. The choice of input device

A menu display on a terminal screen was chosen as the simplest way to present the options available to the user. It is only necessary to 'point' to an option, 'play' for example, in order to select it. The options considered were touch screen, light-pen, graphics tablet, mouse and tracker-ball. Touch-screens and light-pens allow the operator literally to point at menu options. Thus they are easy to learn to use, but would probably be tiring if used for long periods as the operator's arm is unsupported. Touch-sensitive screens offer comparatively poor resolution, limiting the packing density of the options displayed on the menu. Graphics tablets use a special pen (stylus) on a separate horizontal pad, allowing as much precision as a light pen, but with limited data entry features. A tracker ball offers adequate arm support but would need a lot of spinning to travel the full dimensions of the menu with the required precision.

The mouse was chosen because the arm is supported and remains confortable over long periods. It is also easy to position quickly and accurately. If the menu is complex then the mouse has an extra advantage: it has three key-switches, which may be used for selection and for incrementing/decrementing a number, for example.

11.1.2. Control panel

A current development is a control panel to provide close interaction with the replay of audio, for example, to perform 'rock-and-roll' and to adjust gain, both vital features of an editor. A large wheel mimics a tape spool and is connected to an optical encoder to control the speed of replay. Real-time software is being developed to service the control panel.

11.1.3. The menus

There are three menu pages, one for each of the modes of the editor: Edit, User, Directory. The menus allow the user flexibility in working, there being no strict sequence of data entry to follow. Default values are displayed initially and only data differing from these default values need be entered. This can significantly reduce the total amount of information to be entered.

Menu contents are a compromise between the desire to keep page changing to a minimum and the attraction of keeping each menu as simple and uncluttered as it can be. The simpler the menu, the easier it is to learn and use without mistakes.

Block graphics form the line-work framing and partitioning of each of the menus. High-quality characters are used (7 x 12 dot matrix), to improve readibility and reduce eye-strain during long editing sessions. The chosen terminal has a green, short-persistence phosphor, preferred for the same reasons.

11.2. Machine interpretation of menus

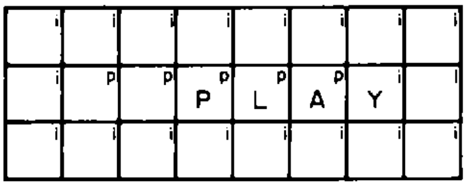
11.2.1. Responding to commands

The roaming of a mouse over a horizontal area produces a corresponding motion of a flashing cursor on the menu. When one of the menu options is selected (by moving the cursor under it and pressing one of the mouse keys), the software must interpret the required action. It does this by referring to a map for the current menu. The map is an 80 x 24 array of codes. There is a unique code for each option and a fixed code to represent a blank or inactive area. Fig. 6 shows a small area of the map (boxes containing a small letter) overlaid on to the corresponding area of the display itself. The letter 'p' is the code for 'play' and the letter 'i' is that for inert. The distribution of 'p' shows where the cursor must be placed to select 'play'.

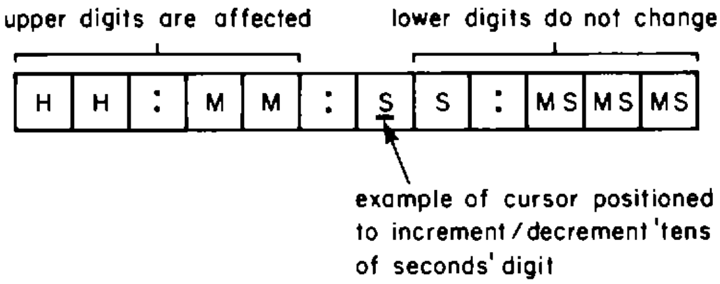
Once an option has been recognised a flashing flag (*) is placed to the left of the option's text on the menu. Then control is passed to the appropriate software function. When the function has finished, the flashing flag is cancelled.

11.2.2. Time-code changes

Some of the menus have boxes containing time-code, showing starting or finishing times for edit commands. Fig. 6 shows such a box. The times may be altered by first pointing to the appropriate box. To acknowledge the request, the editor will brighten the entire box. At this point the cursor may be moved to a particular digit. Once in position the right/left mouse keys cause the digit to be incremented/decremented, with digits to the left (more significant digits) affected by carrying or borrowing. The digits are seen to 'crank' either up or down, like a car's odometer. If the key is held pressed, the cranking is slow at first but speeds up to about ten increments/decrements per second whilst the key



Overlay of the map on the menu.



A time-code box.

key	H	hours
	M	minutes
	S	seconds
	MS	millisecond

Fig. 6 – Menu interface techniques.

remains pressed. The slow initial rate is necessary to allow a short press of the key to cause only a single increment/decrement. When the alteration is complete, the cursor must be moved outside the box. This cues the program to check the validity of the time-code and clean up the display by suppressing leading zeros. Then the mouse is free to roam and select another function.

12. System configuration

12.1. Interfacing to recording devices

RIO and the disc can be considered a sub-system acting as a demand-fed buffer (Fig. 4). In the current equipment, the data transfer bus is the proprietary DEC Qbus and this must be shared with normal bus use for running programs etc. A commercially available disc controller provides a data path to a Storage Module Drive (SMD) with 68Mbytes formatted capacity[9].

More recent developments in computer peripherals have led to a separation of the data transfer bus from the system bus. This will be adopted so that audio transfers can be carried out without penalty to the running of programs, and also so that transfers between recording devices can be done directly. Relatively low cost 5¼" Winchester discs now have capacities of 190Mbyte with acceptable transfer rates. Optical discs and streaming tape cartridges are also available in compatible formats.

For the purposes of this work, it is assumed that progress will continue in this area. Of more direct interest is the means for interfacing with studio equipment and the provision of advanced editing features.

12.2. Internal bus systems

To evaluate studio equipment and the man-machine interface, a comprehensive architecture has been defined, as shown in Fig. 7. The figure concentrates on the input/output arrangements of the editor for audio and control.

There are four buses, each with differing functions.

a) AUDbus: This time-multiplexed bus routes data from analogue or digital sources to the varispeed processor (VSP) and auxiliary data formatter (ADF).

b) RVbus: The RVbus passes a bi-directional time-multiplex of audio and pre-formatted auxiliary data. The data rate of the bus varies proportionally with the replay speed.

c) SYNCbus: The precise interaction between control wheel, faders, etc and the formatting and processing in RIO and the VSP demand a direct synchronous link. If the data transfers were handled directly by the system processor, the speed of response would be greatly reduced. The bus guarantees virtually instant response and a steady update rate so that gain and speed control vary smoothly.

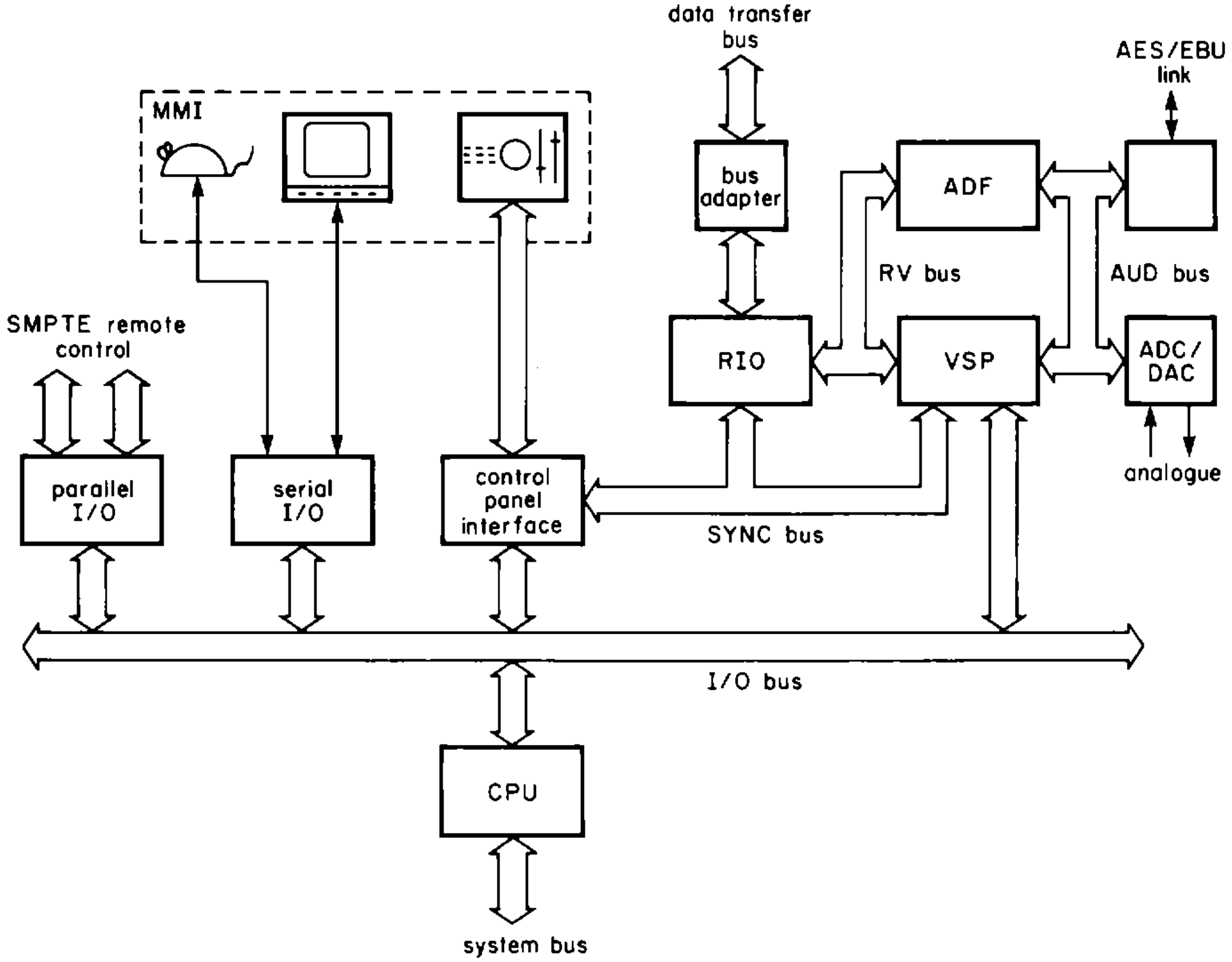


Fig. 7 — Internal bus system of the editor.

d) I/Obus: The overall control and monitoring of the special purpose hardware is managed at relatively low data rates using a proprietary bus. The system has access to all the control information and status of the various units without the overhead of interrupts at the rate of the SYNCbus. Connection to other devices for synchronisation and SMPTE remote control can be made using standard proprietary modules which interface to this bus.

13. Software development

13.1. Operating systems

The editing software runs under the supervision of the operating system, Idris.™ This operating system is very similar to Unix™ [20]. Idris services the editor's inputs and outputs, in addition to permitting control of RIO. The main advantage of Unix™ (Idris™) is that it offers a good environment for the development of software. It has a wide range of simple commands. A combination of these can be easily linked to form a powerful custom-command. Custom commands save time where a long, complex sequence of standard commands is needed frequently.

Unix ™ organises its files in a tree-like structure, with a primary directory at the top of the tree. The primary directory (called the root) contains sub-directories, which themselves contain directories. Fig. 8 gives a simple example of the organisation within a sub-directory. A large number of files is easily managed in this way. Editing software takes advantage of the directory structure to classify its many functions. Thus functions in the same directory will be related and the directory's level will reflect the level at which the functions work. For example, functions that link the editor to the pieces of hardware such RIO, will be at the bottom of the tree, whilst functions that supervise the software are at the top. This

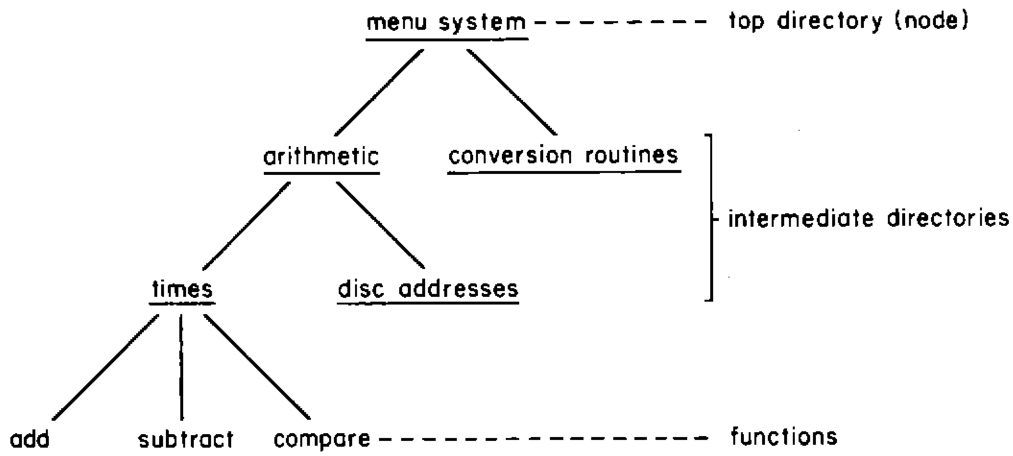


Fig. 8 – Edit functions classified using the Unix directory system.

ability to classify functions fits well with 'top-down design', discussed later. Appropriate directory names help to locate functions, e.g. a directory called 'mouse-interface' will contain functions for just that.

Within Unix are a large number of services that the editing program can call on. For example, a directory of audio files can be sorted by a call to the operating system. Also the ability to start another program, from within a program, has been useful in linking software modules together at an early stage of development.

To allow precise control of audio replay, the operating system must handle frequent signals from devices such as RIO, the control panel, the mouse, etc. Unix is not only slow to respond to these signals but it fails to guarantee that such signals can be serviced in a specified time.

By contrast, a real-time operating system can respond quickly to external events as they occur, and offers a predictable response time. For this reason, current and future software development will use the Unix environment for development and non-real-time testing and use a real-time operating system for time-critical debugging and final operation in the target environment.

13.2. Designing the software

The software has to weld all the items of hardware together with its own structure to form one machine, the Digital Audio Editor. It assists the process of locating and storing edits, and permits the remote control of devices such as the DATR. By displaying information on menus, and allowing input via the mouse, it simplifies the control of the editor. The user is left to define only essential features of an edit, whilst the software calculates the machine-dependent details.

In accordance with 'top-down design' the large complex task was split into smaller independent functions. Each of these functions has responsibility for one key aspect of the editor and each of them can call on a large number of lower-level functions. Because many of the functions are designed for just one simple purpose, they are re-used many times. To show the hierarchy of functions, a series of concentric shells are used, with high-level functions occupying outer shells, and lower, more machine-dependent functions towards the centre (Fig. 9).

13.2.1. Structured analysis

Careful splitting of the software allows smaller units to be developed separately. However, before the split in development, the data interface between functions must be well defined, by writing a 'data dictionary'. This is done by analysing the types of data, data structures, and disc files that the system needs and is useful for rationalising data flow within the system. Once complete, it is a standard, which all the various functions have to observe. Fig. 10 lists the types of data used by the editor.

With these techniques, the software can be produced in the form of modules: groups of associated functions. An example of the module is the collection of mouse software, which could be easily replaced with a light-pen module.

The methods of structured analysis[21,22,23,24] aid the top-down design process by producing a plan of how software elements talk to each other (by passing data and/or control signals). With this information, the software partitioning can be optimised, as can the packets of data that pass between the partitions.

Structuring and lucid naming within the software help its readability. Better understanding

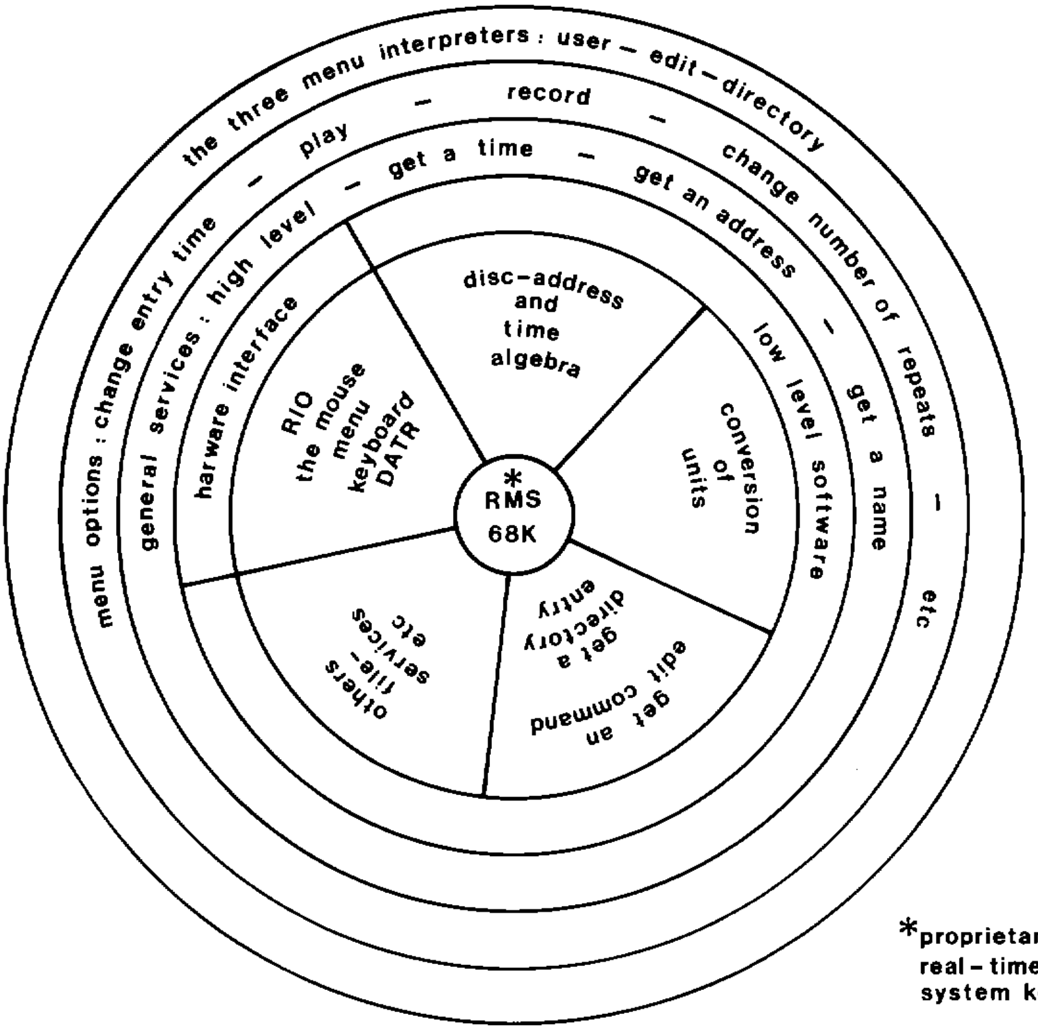


Fig. 9 – Function hierarchy in the editing software.

follows, and the processes of debugging and enhancement become faster and easier. The software's useful life is lengthened because it is easier to alter when the hardware is updated.

13.2.2. Control and data flow

A structured analysis helps to decide the key elements of software at the earliest stage of design. Fig. 11 is a simplified version of the result of this analysis on the digital editor. It maps the control and data flow through the system. Starting on the left, the ultimate source of data and control is the user working through the mouse and keyboard (and control panel under development).

Lozenge 1 : the menu interpreter : there is one per menu. It decides which option has been selected, if any, and choses the corresponding function. Then it passes control to that function and waits to receive control back when the function has finished. The interpreters do not handle edit data.

Lozenge 2 : the selected menu function : receives control from the interpreter and becomes master of all the system resources until it finishes. Data concerning edits and audio files are received direct from the user and are stored in edit files, i.e. files which contain a list of edit decisions. The function can send information to the user via the menu display. Some functions, such as 'play', take data from both an edit file and the audio directory, then call a subsidiary function (see Loz. 5) to initiate and maintain a faster transfer of audio from the disc to RIO. Any problems are referred to an 'exception' (error) manager, for example if a file cannot be found or there is a hardware fault.

Lozenge 3 : the display manager : is notional because although there are many small

edit data

Edit commands	: contain vital information about edits
Joins	: contain splice details (subset of Edit commands)
Edit files	: collections of edit commands

audio directory data

Audio Directory Entry	: specifies name, time of recording, sampling frequency etc.
Audio Directory	: collection of audio directory entries

menu data

Text	: specifies position, number of character etc.
Graphics	: specifies position of a graphical character
Boxes	: containing time code and changeable numbers

Fig. 10 — Examples of data structures within the editor.

functions designed to make the menus easy to update, there is no overall manager. If written, this would make the display software less dependent on other software.

Lozenge 4 : error handling : at present, the 'exception' manager puts a warning on the menu but allows the function to continue. For a finished system, this would be a major area for development.

Lozenge 5 : audio transfer functions : the fast audio transfers between RIO and the Winchester disc are initiated by this software. It interprets edit commands and reformats the data into machine-level parameters.

Lozenge 6 : remote control functions : a package of functions are available for remotely-controlling a DATR, enabling the editor to supervise *dubbing from tape to disc etc.* These functions like those in Loz. 5 are machine-dependent and would need changing if a different DATR is used.

Lozenge 7 : label retrieval : when audio is replayed, labels are available as a secondary output. This function receives labels and formats them for

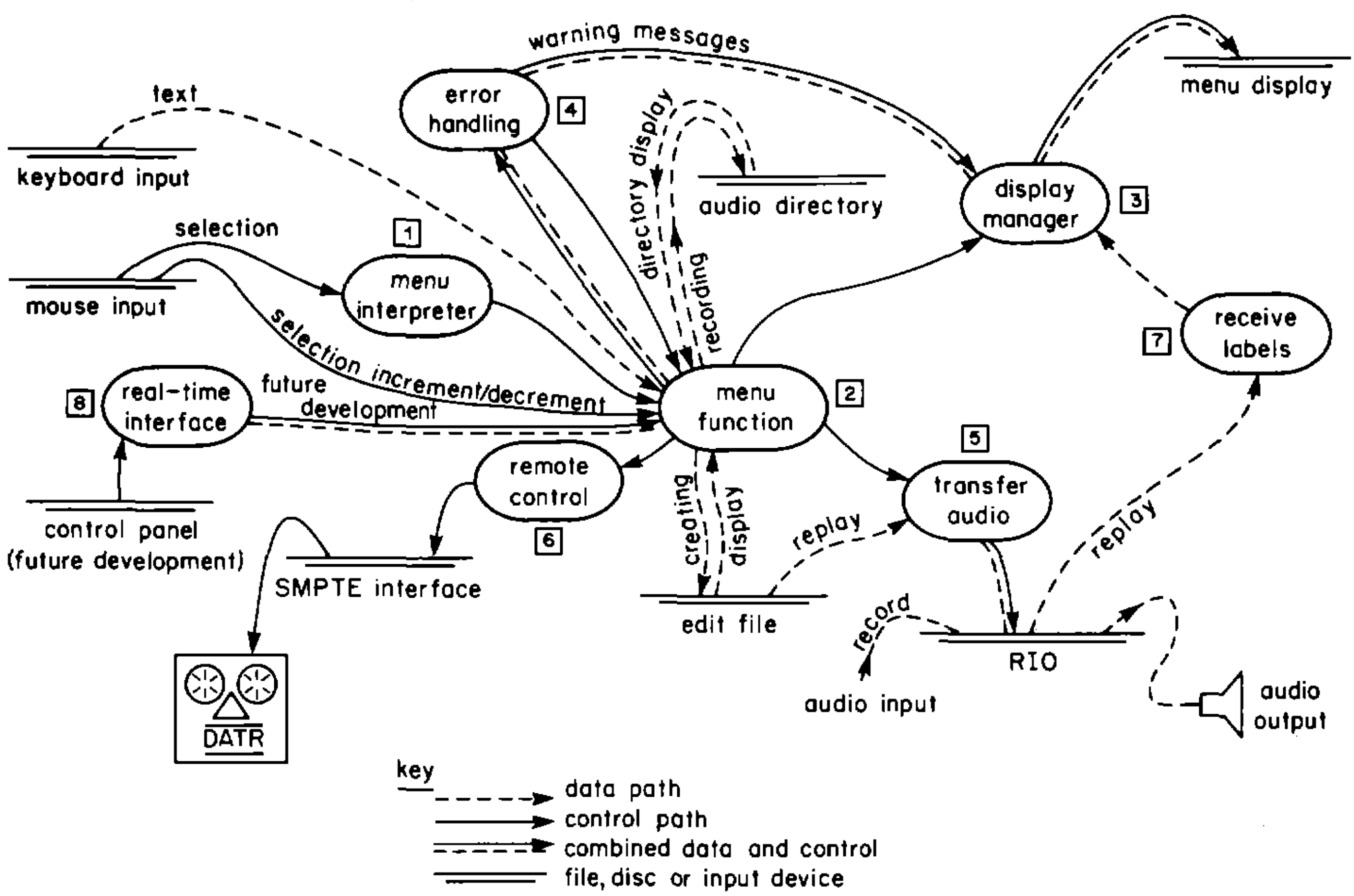


Fig. 11 — Control and data flow within the editor.

display, either on the menu, or a separate display. It runs on a separate processor, outside the Unix operating system. In this way it can respond quickly and reliably (one label is received every 4 ms).

Lozenge 8 : the future – real time control of digital audio : the previous lozenge (labels) marks the start of an offshoot from the current editor. It is an early stage of a real time system. The shoot will eventually grow to become a new editor, independent of the existing system. Development of software for reacting to the control panel is maintaining this growth.

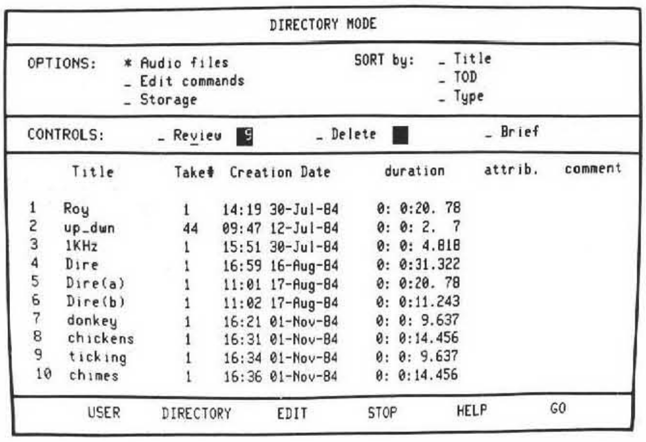
13.3. Implementation

All of the software is in the 'C' language, which, like Pascal, promotes structured programming. Each of the elements defined by the design becomes a 'C' function and is kept in a file bearing the same name. As mentioned, functions are grouped into directories, whose hierarchy is arranged to reflect the hierarchy of functions within the software. Vital parameters of the system are sorted into files, each file having a common theme.

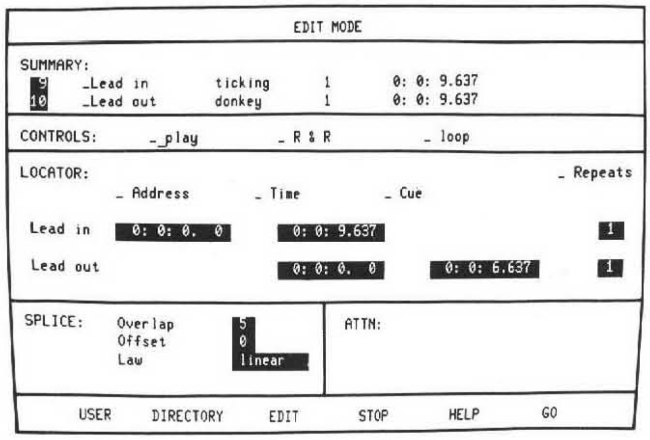
Custom commands were mentioned in connection with Unix™. They can be constructed to form software 'tools' which speed up development. An example of a tool is a program which collects together all the functions and 'builds' the editing system from them. This automation saves time after modifications have been made to individual functions. Many tools are provided with Unix,™ e.g. tools to help document the software, to check the program for bad style and keep track of changes.

14. Experience to date

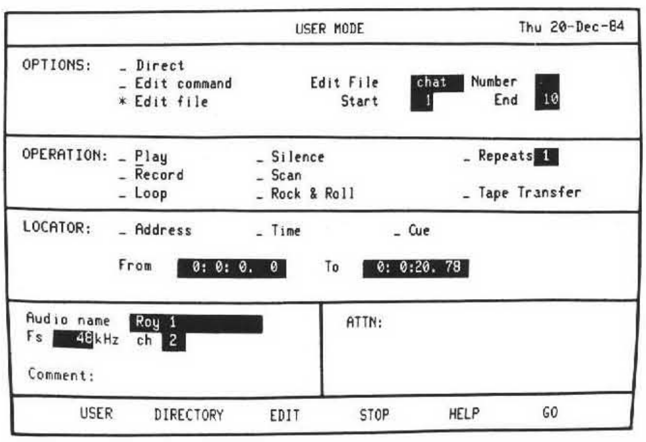
An editing system has been built as in Fig. 11 with the 'real-time' features of the VSP and control panel under construction – Fig. 12d. Stereo data transfers have met the expected performance described in Section 7 and the auxiliary data capacity has been used to print the lyrics of a song while it



(a)



(b)



(c)

(d)

Fig. 12 – The experimental editor.

a) a 'directory menu' b) an 'edit menu' c) a 'user menu' d) the operator's control unit under development.

is being played. Considerable effort has been devoted to the man-machine interface and meaningful feedback is being obtained from operators.

The system runs under the Idris™ operating system with some 15000 lines of 'C' programming. Three menus are available. The first provides directory information of audio files (Fig. 12a) and edit commands. A second is used to create and rehearse edits (Fig. 12b) and a third menu allows the user to treat the disc like a very intelligent tape recorder (Fig. 12c). No keyboard skills are required since nearly all data entry is via the three buttons of the mouse.

Initial reactions have been favourable and are spurring the addition of real-time features. The original prototypal system is being rebuilt as a development system plus a target system. The development system is a Unix™ System V workstation, while the target runs Versados,™ a real-time multi-tasking operating system.

15. Conclusions

A digital audio editor has been constructed and is still under development. Its specification has been the result of a number of investigations into current audio editing practice, the current progress in digital audio recording and studio interfacing, the need for carefully organised software development and the necessity for excellent man-machine interfacing.

A hierarchy of editing techniques have been identified, in which tape-cut editing, a new method of 'jump' editing and the random access features of discs all play a part. At the top-level of the hierarchy, only a disc is required for editing and new facilities based on auxiliary data are outlined. These will significantly improve efficiency in edit point location and provide the tools for high accuracy and repeatable edits.

16. Acknowledgements

This work was carried out under a collaborative agreement with Willi Studer AG.

17. References

1. RUMSEY, F. 1984. Sound dubbing techniques for film and video : a comparison. BKSTS Journal. 1984, **66**, 6, pp 272-277.

2. EDEN-EADON, J. 1977. Sypher, a modern method of dubbing on videotape. BKSTS Journal. 1977. **59**, pp 306.

3. DAVIS, D.E., YOUNGQUIST, R. 1979. Electronic editing of digital audio programmes. International Conference on Video and Data Recording. Southampton, U.K. 1979. pp 385-393.

4. OHTSUKI, T., KAZAMI, S., WATARI, M., TANAKA, M., DOI, T.T. 1981. Digital audio editor. 68th AES Convention, Hamburg, 1981. Pre-print 1743(B-5).

5. TANAKA, K., ONISHI, K., KAWABATA, M. 1979. On Tape-cut editing with a fixed head PCM tape recorder. IEEE Trans. ASSP, 1979, *ASSP-27*, 6, pp. 739-745.

6. KOWALSKI, M.J., GLASSNER, A. 1982. The N.Y.I.T. digital sound editor. Computer Music Journal, 1982, **6**, 1, pp 66-72.

7. INGEBRETSEN, R.B., STOCKHAM, T.G. 1984. Random access editing of digital audio. Journal AES, 1984, **32**, 3, pp 114-122.

8. GRIFFITHS, M., BLOOM, P.J. 1982. A flexible digital sound editing program for minicomputer systems. Journal AES, 1982, **30**, 3, pp. 127-134.

9. McNALLY, G.W., GASKELL, P.S. 1984. Editing digital audio. Proceedings ICASSP-84, San Diego, 1984, pp. 12B4.1-4.4.

10. LAGADEC, R. 1984. New concepts in digital audio editing. 75th AES Convention, Paris. Pre-print 2096(H-6).

11. WHITE, R.M. 1983. Magnetic disks: storage densities on the rise. IEEE Spectrum. August 1983, pp. 32-38.

12. LLOYD, P. 1983. Optical disc drives – the break-through in data storage. Electronics and Power. October 1983, pp. 711-714.

13. ITOH, T., OHTA, T., SOHMA, Y. 1980. Real-time transmission system of commercial messages in radio broadcasting. 67th AES Convention, New York. 1980. Pre-print 1682(H-1).

14. MCNALLY, G.W. 1984. Varispeed replay of digital audio with constant output sampling rate. 76th AES Convention, New York. Pre-print 2137(A-9).

15. GASKELL, P.S. 1985. A hybrid approach to variable speed replay of audio. 77th AES Convention, Hamburg. 1985. Pre-print 2202(B-1).

16. LAGADEC, R., McNALLY G.W. 1983. Labels and their formatting in digital audio recording and transmission. 74th AES Convention, New York. 1983. Pre-print 2003(A-5).

17. McNALLY, G.W. 1985. Fast edit point location and cueing in disc based digital audio editing. 78th AES Convention, Anaheim. 1985. Pre-print 2232(D-10).

18. ABBOTT, C. 1984. Efficient editing of digital sound on disc. Journal AES, 1984, **32**, 6. pp 394-402.

19. LAGADEC, R., DOI. T.T. 1982. A digital interface for the interconnection of digital audio equipment. 71st AES Convention, Montreux, 1982. Pre-print 1883(G-6).

20. KERNIGHAN, B.W., PIKE, R. 1984. The Unix Programming Environment. Prentice-Hall. New Jersey. 1984.

21. KEITH, N.L. 1984. Software Tools Automate Structured Analysis (article). Electronics Week, August 20th 1984.

22. DeMARCO, T. 1978. Structured analysis and system specification. Yourdon Press, New York, 1978.

23. GANE, C., SARSON, T. 1979. Structured systems analysis : Tools and Techniques. Prentice-Hall, Englewood Cliffs, New Jersey. 1979.

24. McMENANIN, S., PALMER, J. 1983. Essential systems analysis, Yourdon Press. New York. 1983.

GLOSSARY

Buffer :	Computer memory for holding data temporarily between processes.
Cylinder :	In a disc system a combination of tracks, on different surfaces at the same radial distance.
Chequer-boarding :	A condition resulting from repeated storage and erasure of different parts of a disc which results in data files becoming fragmented.
Direct Memory Access (DMA) :	A mechanism in a computer that bypasses the central processing unit to gain access to memory. It is often used when large blocks of data are transferred from memory to a peripheral.
Disc Format :	A pattern of sectors created by a formatting program before the disc is used, consisting of the header information and null data written at the desired physical positions on the disc.
Latency time :	In a disc system, the time taken for the required data to be rotated to reach the read/write head. Worst case delay is one revolution.
Mouse :	A small hand-held device which translates hand movements into cursor movements on a screen. The mouse may also have one or more buttons for making selections based on the screen display.
Multi-tasking :	The simultaneous execution of two or more applications programs in a computer.
Operating System :	A computer program that performs basic operations such as governing the allocation of memory, accepting interrupts from peripherals, and opening and closing files.
Pipe-lining :	A method by which several computations are carried out simultaneously in the manner of an assembly line.
Ratio Addressing :	A technique by which a number of contiguous files can be read at different rates without address rounding error.
Sector :	A block of data on a disc, conventionally 512 or 1024 bytes with identifying header information and error correction data.
Seek Time :	In a disc system, the time taken for the head to move from one cylinder to the required cylinder including mechanical settling.
Spiralling : :	The ability to read a disc made up of discrete tracks and cylinders as if it were a single continuous spiral track.
Skew Sectoring :	A method by which the first sector of adjacent tracks and/or cylinders on a disc is off-set to compensate for the track-to-track seek time and so minimise latency time.
Track :	A sequence of sectors making up one complete revolution of the disc on a single surface.
Winchester disc :	A magnetic disc storage technology in which the discs operate in a sealed, clean air environment.

Unix is a trademark of Bell Laboratories :
VERSAdos, RMS68K are trademarks of Motorola Inc. :
Idris is a trademark of Whitesmiths Ltd.

(PH-274)

Printed by BBC RESEARCH DEPARTMENT, Kingswood Warren, Tadworth, Surrey, KT20 6NP